IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. No.	:	10/758,368
Applicant	:	Simon C. Steely Jr.
Filed	:	January 15, 2004
Title	:	SYSTEM AND METHOD FOR UPDATING OWNER PREDICTORS
TC/A.U.	:	2185
Examiner	:	Midys Rojas
Docket No.	:	200313752-1
Customer No.	:	022879

Commissioner of Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION OF SIMON C. STEELY JR. PURSUANT TO 37 C.F.R. § 1.131

Dear Sir:

1. I am a co-inventor of the apparatus disclosed in U.S. Patent Application No. 10/758,368 (the "'368 application").

2. I conceived the invention presently claimed in the '368 application with co-inventor Gregory Tierney prior to January 13, 2004, as evidenced by the invention disclosure describing the claimed invention that we submitted on June 13, 2003 to Hewlett Packard's legal department (see Exhibit A).

3. Indeed, we conceived the claimed invention prior to June 13, 2003.

4. The declarant further states that the above statements were made with the knowledge that willful false statements and the like are punishable by fine and/or imprisonment,

or both, under Section 1001 of Title 18 of the United States Code, and that any such willful false statement may jeopardize the validity of this application or any patent resulting therefrom.

Date: October 5, 2009

Simon C. Steely Jr.

Exhibit A

(14 PAGES INCLUDING THIS COVER PAGE)

Enter PD Number

HEWLETT PACKARD

INVENTION DISCLOSURE

DATE RCVD: 06/13/2003

Tag Number: T0004497 PDNO: 200313752 ATTORNEY: LPG

All IPG inventors and other inventors who have access to Disclose should submit their disclosures through that application. The url is < https://wkrpweb1.cv.hp.com/dbi?application=disclose >. Invention Disclosures submitted here by inventors who have access to Disclose will not be processed at all.

Instructions: *The information contained in this document is **HP CONFIDENTIAL** and may not be disclosed to others without prior authorization. Submit this disclosure to the HP Legal Department as soon as possible. No patent protection is possible until a patent application is authorized, prepared, and submitted to the Government.*

Red text indicates a required field.

Descriptive Title of Invention: Owner Prediction with Processor-side Directory Caches in a Distributed cc-NUMA system.
Name of Project: Windjammer
Product Name or Number:
Submitter Location (City): Other
Was a description of the invention published, or are you planning to publish? If so, the date(s) and publication(s):
No
Was a product or prototype including the invention (i) announced, offered for sale, or sold to any third party (for example, customer, supplier, contract manufacturer), or (ii) sold to HP by, for example, a supplier or contract manufacturer, or (iii) is such activity proposed? If so, when and to whom?:
No
Was the invention disclosed to anyone outside of HP, or will such disclosure occur? If so, the date(s) and name(s):
No
If any of the above situations will occur within 3 months, call your IP Attorney or the Legal Department now at 1-898-4919 or 970-898-4919
Was the invention described in a lab book or other record? If so, please identify (lab book #, etc.)
No
Was the invention built or tested? If so, the date:
No
Was this invention made under a government contract? If so, the agency and contract number:
No
Description of Invention: Please describe your invention in detail using the following outline. A. Prior solutions and their disadvantages (attach copies of any pertinent product literature, technical articles, patents, etc.). B. Problems solved by the invention. C. Advantages of the invention over what has been done before. D. Description of the construction and operation of the invention. (include appropriate schematic, block & timing diagrams, drawings, samples, graphs, flowcharts, computer listings, etc.). Electronic Attachment
List any pertinent patents material to the invention.
List any articles or references or devices pertinent to the invention.

Identify Inventor(s): Pursuant to my (our) employment agreement, I (we) submit this disclosure on this date: 06/13/2003

Employee No. 4719590	Name: Simon C. Steely, Jr.	Telnet: 508-467-4631	Mailstop: MRO1-1/P5	Entity & Lab Name: HPSL - MRO section
Employee No.	Name: Gregory	Telnet: 508-467-4499	Mailstop: MRO1-1/P5	Entity & Lab Name:

10329582	Edward Tierney			HPSL - MRO section
Employee No.	Name:	Telnet:	Mailstop:	Entity & Lab Name:

Identify Witness(es): *(It's best to identify the person(s) to whom invention was first disclosed.)*				
The invention was first explained to, and understood by, the witness(es) on this date: May, 2003				
Name: Steve Van Doren	Employee No.	Telnet:	Mailstop:	Entity:
Name:	Employee No.	Telnet:	Mailstop:	Entity:

Inventor & Home Address Information:		
First Inventor's Full Name: Simon C. Steely, Jr.		Citizenship: U.S.A.
Street 8 Anna Louise Dr		
City Hudson	State NH	Zip 03051
Do you have a Residential P.O. Address? No	Description	
Second Inventor's Full Name: Gregory Edward Tierney		Citizenship: U.S.A.
Street 161 Boston Rd		
City Chelmsford	State MA	Zip 01824
Do you have a Residential P.O. Address?	Description	
Third Inventor's Full Name:		Citizenship:
Street		
City	State	Zip
Do you have a Residential P.O. Address?	Description	

Hardcopy Files:

owner prediction.zip

Owner Prediction with Processor-side directory caches in a Distributed cc-NUMA system.

The Invention

See co-pending invention disclosure entitled, "Owner-Speculative Scalar Protocol for Distributed cc-NUMA" for an example of a protocol that supports owner prediction. See the paper, "Owner Prediction for Accelerating Cache-to-Cache Transfer Misses in a cc-NUMA Architecture", by Manuel E. Acacio, et al for an example of another approach.

Owner prediction is the process of guessing the third-party cache location of the coherent version of the data. A system using owner prediction will send a request packet directly from the requesting node to the third-party cache and return the data in 2-hops. A protocol that sends its requests only to the home node would require 3 to 4 hops to retrieve data from a third-party cache..

This invention is an approach to performing owner prediction that works well. We've done performance simulations for this approach as part of our Windjammer investigation work. See file procside_cacheing.ppt included with this disclosure.

A Directory Cache is a structure that is in front of each directory and remembers the recent entries accessed in the directory. The structure is built out of cache technology that allows it to be accessed much faster than accesses to the actual directory. This invention is concerned only with the directory accesses that change the owner of the line. Our simulations have shown that modest sized directory-caches have high hit rates on blocks that are owned by a remote cache.

This invention places structures that behave similar to a directory cache at each processor node. The structure is organized to identify blocks that are owned by a third-party cache. The contents of the structure are manipulated with system commands from the directory controller. For each directory access that results in a change to the ownership, a message containing the address and new owner of the line is broadcast to each of the owner-prediction structures.

As a miss comes out of the processor, it examines the owner-prediction structure. If a match is found it proceeds to obtain the data from the third-party owner. Note, in using the OSSP protocol (mentioned in co-pending disclosure) request messages will be sent to both the third-party and the home-directory in parallel. The protocol does not require that the predicted owner is actually able to service the request. Thus, the owner-prediction structure need not be meticulously maintained to avoid race conditions that may develop while ownership state changes.

An alternative embodiment could be included to work with co-pending disclosure entitled, "Mechanism Using Coherent Signal to Validate Eager Reply from Shared Copy

Located in Neighbor Cache" to recognize and broadcast sharing update information to appropriate close neighbors as well. This would make the identification of shared copies highly accurate as well.

Another alternative embodiment would be to combine this processor-side, cache-like owner predictor with a conventional pattern-based owner predictor. The processor-side directory-change updated predictor works well when the ownership change was timely (happened long enough ago to allow the network latency of the update to be buried). However, when the ownership change has happened very recently (this miss occurs in the shadow of an update to the processor-side directory-cache predictor) then the pattern-based predictor can help out by identifying some target nodes to guess on where the data is. A pattern-based predictor could also be used to better manage the contents of the processor-side predictor, for example by assisting the replacement policy of the owner-cache to improve the hit rate of misses that are most critical to the processor.

Advantages

This prediction approach is more precise than the pattern-matching strategies of other known approaches, because its use of system update messages provide the true ownership state of a line. This yields the following advantages:

- Able to record only a single owner, whereas less precise approaches often require several parallel guesses. Our predictor is thus able to hold more blocks in the same amount of storage space than is required by an approach that records multiple ownership targets.
- Only need to store state for those blocks that may be cached at a third-party. If an address is not found in the predictor, it can assume that the data can be found in memory and that speculation is not beneficial.
- State updates are more timely. Most cache-to-cache transfers occur after a block has been written, which is often identified in our system with an ownership change. Alternatively, pattern-based predictors are trained by the prior miss and/or prior probe. Timely updates increase the likelihood that a third-party target can be accurately identified by our predictor.

Problems Solved

Identifying the location of remotely cached data reduces the effective latency of accesses to the system.

Technology trends are such that link bandwidth between nodes in a distributed system will be higher than the bandwidths of the nodes that are being interconnected. This means that protocols that broadcast information will be more likely to work well since they won't suffer the queueing delays encountered in present networks.

Broadcast of predictor table information from directories to nodes will be less of a handicap as technology improves. The subsequent reduction in effective system latency will result in increased system performance.

Disadvantages of the Past

Earlier approaches, such as the ownership prediction scheme in the Acacio paper reply on information from fills and other responses to train the ownership prediction structure. This works only if there is a pattern to the processor that owns the data. The prediction solution here works even without patterns in the owner of the data.

On-Agent Approaches to minimizing network latencies

Apr 17, 2002

Greg Tierney
Stephen Van Doren
Simon Steely

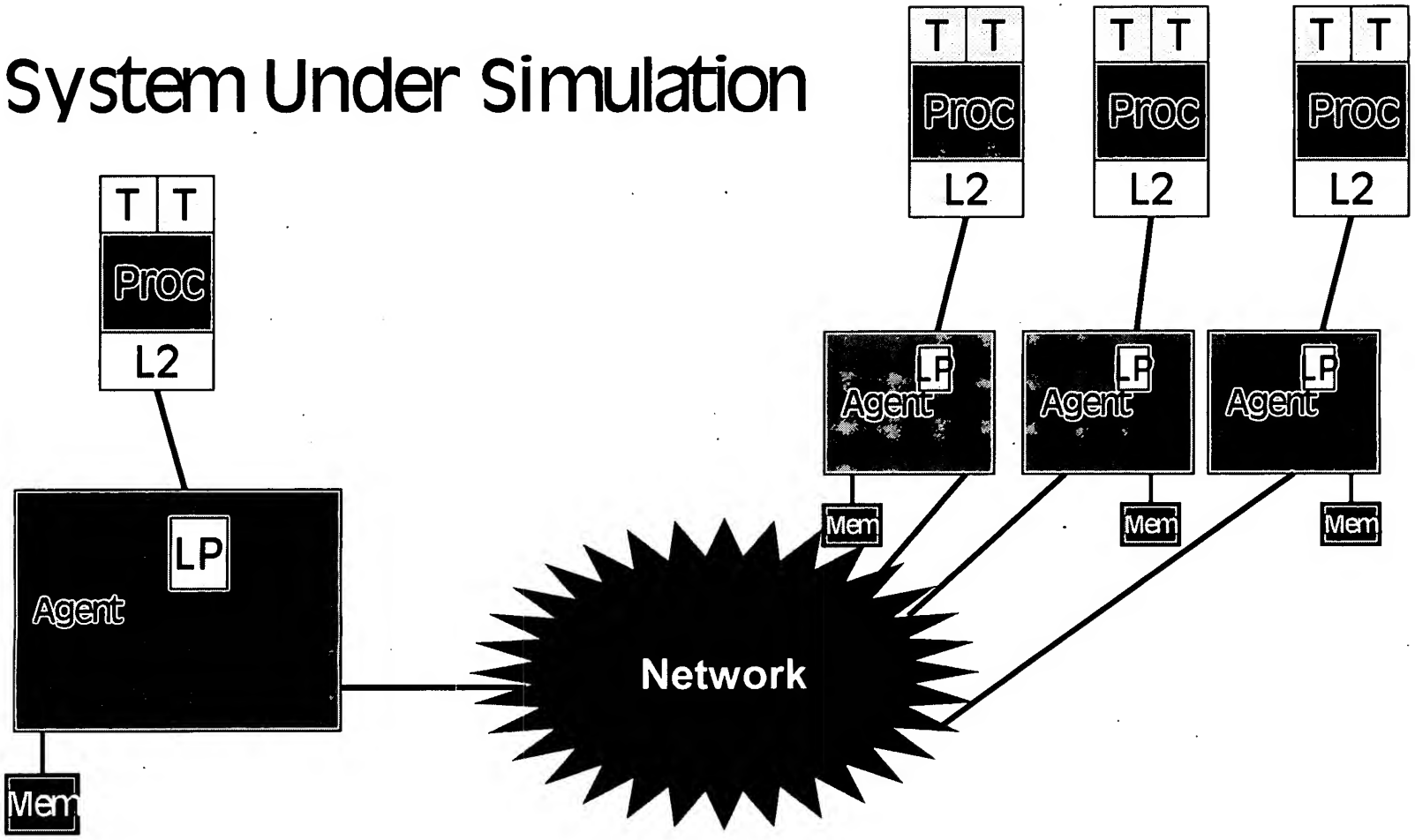
Executive Summary

- Prefetch cache on processor side has problems being timely
- Owner Prediction on processor side (shortening 3-hop misses to 2-hop misses) is timely and has promise.

Caching at Memory-side of network or processor-side of network?

- Consider moving cache at memory to the processor side of the network by sending it across the network and caching on the processor side.
- There are two categories of caches being considered:
  - Data information, whereby the cache is a prefetch structure. Scheme studied here is to fetch a larger line size and leave the extra data in the prefetch structure.
  - Directory information, whereby the cache is an owner prediction structure to convert 3-hop misses to 2-hop misses. (Could also be a shared-copy predictor also).
- Timeliness – Can we get the cache located on the processor side of the network loaded early enough to make it work well.
  - We chose 1000 instructions as our measure of timeliness.

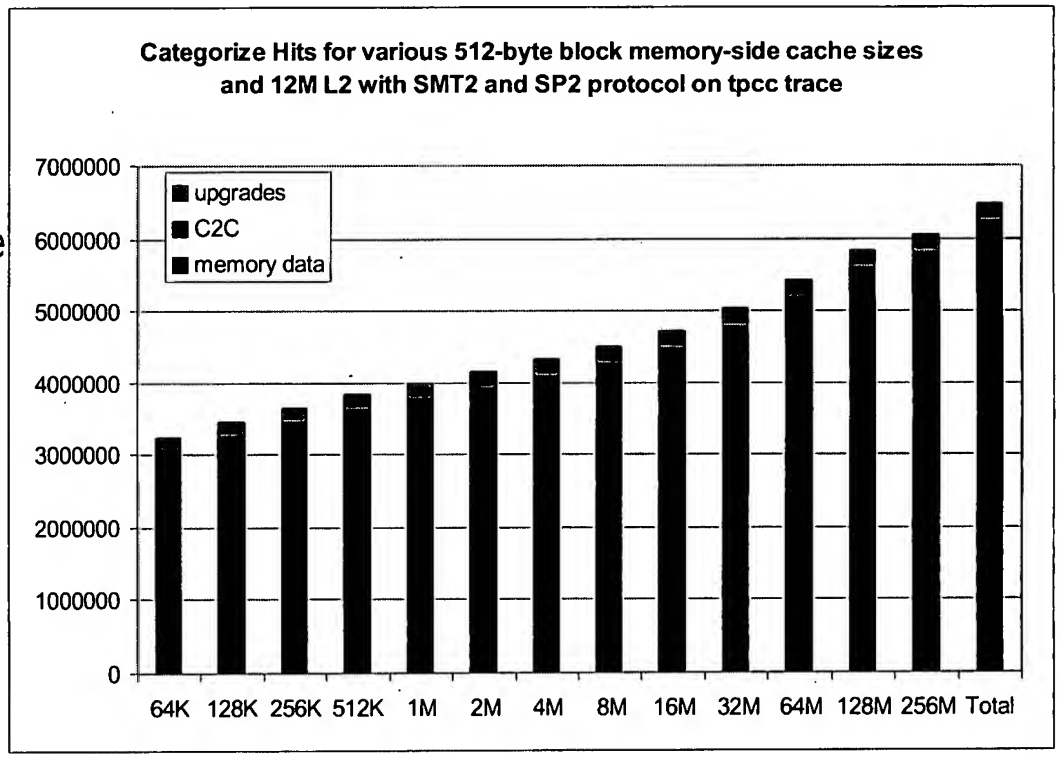
System Under Simulation



- 4 processors, each with two threads.
- On-chip LPs on the processor side of the network.

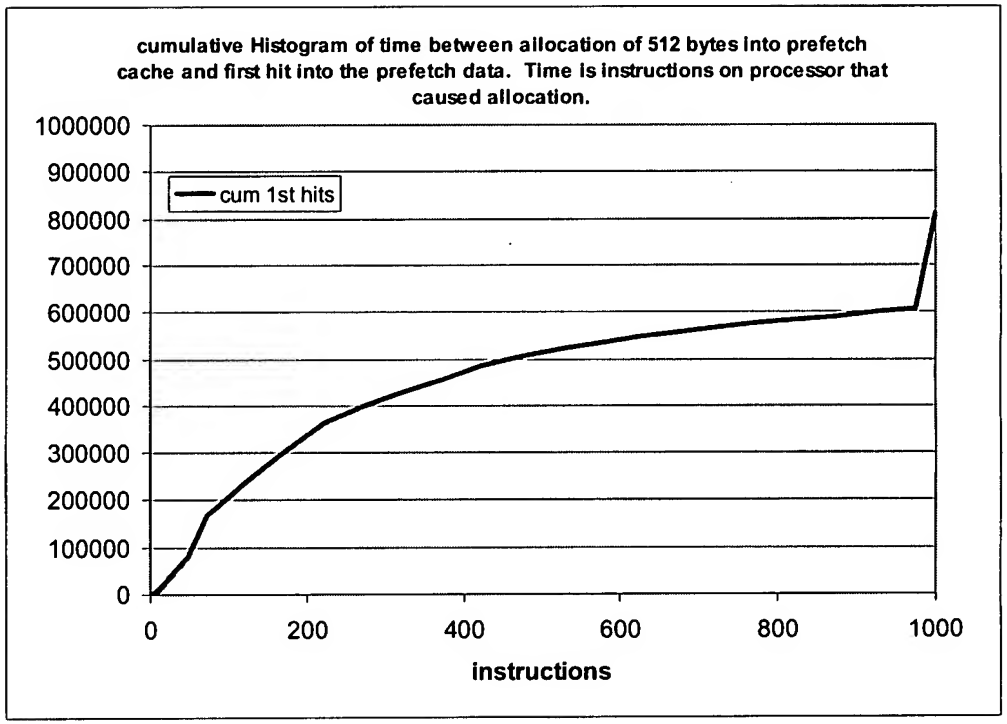
Review of 512-byte block memory-side prefetch Caches

- In summary, the graph indicates that about ½ of the red misses appear to be serviced by a small prefetch structure.
- In discussion with Pat Knebel we wondered if these small prefetch caches could be located across the network on the processor side.
- Using 512 byte prefetch small-size structures would transfer 2x network traffic than without prefetch.
- Largest question is whether the hits to the prefetch data are far enough from the first access to allow the prefetched data to cross the network and actually achieve a performance advantage?



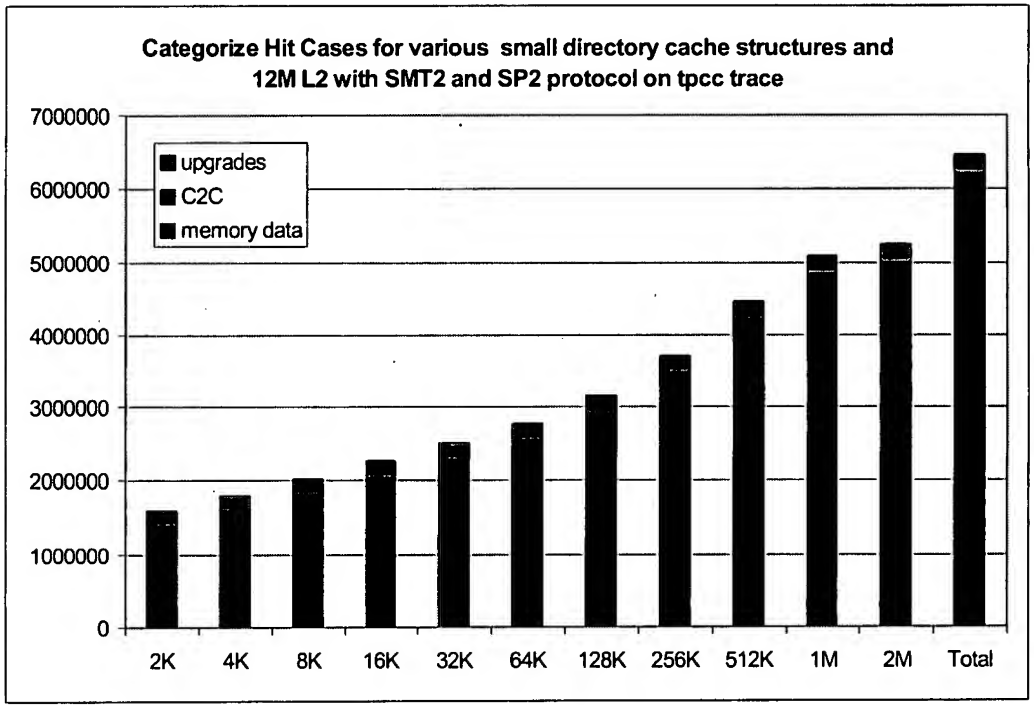
Examining time between allocation and first hit.

- On the chart the 1000 bin represents all gaps of that size and larger.
- For each cache entry that has a hit, there are on average 1.73 hits with the small prefetch caches.
- We'll arbitrarily use 1000 instructions on the processor that caused the allocation as the time unit that states there is enough time to get the data across the network.
- This works out to 350,000 hits would happen in the LP.
- This is out of 3.5 million possible hits, so only a 10% hit rate at the cost of doubling the network bandwidth needs.
- This leaves 40% of the misses as in-flight races between the prefetch and the demand miss which must meet up somewhere along the memory-access pipeline.



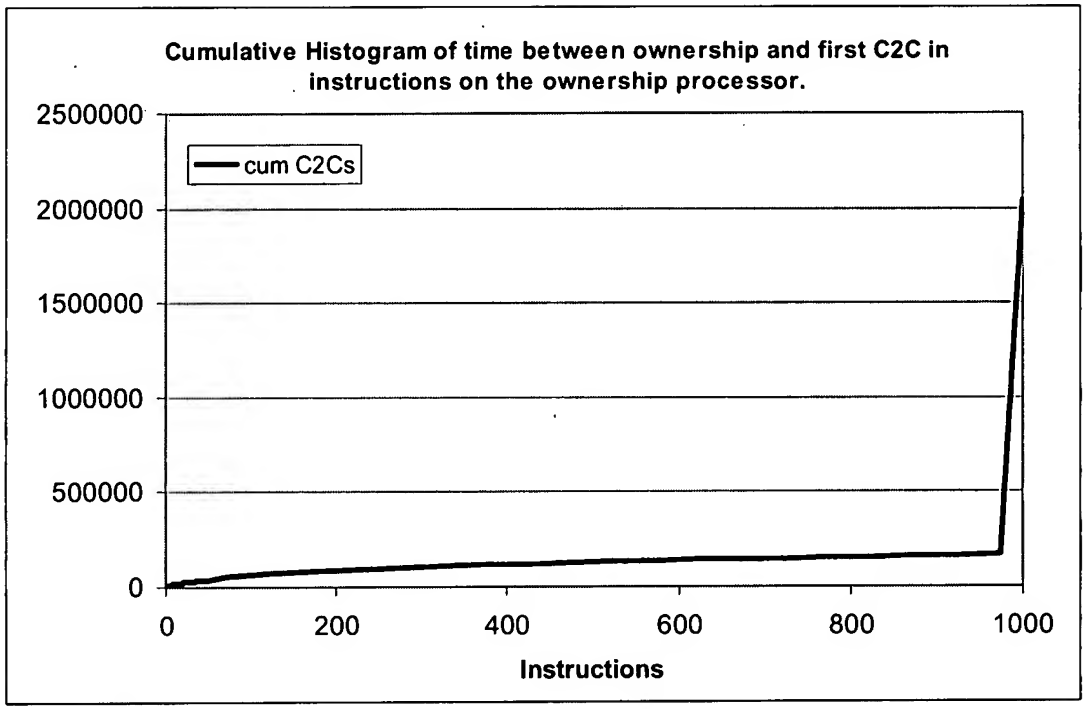
Let's go back and look at conventional small directory cache structures (No prefetch).

- What if we try and push the blue directory information across the network and turn C2C misses into two hops instead of three?
- If we look at a 32K entry structure, it looks like up to 85% of the C2C cases could be improved if we can get the owner information to the LP in a timely manner?



Examining time between ownership and C2C

- 1.87 million of the C2C cases (or 92%) both hit in the 32K entry LP structure and happen more than 1000 instructions after the ownership.
- Out of a total of 2.4 million C2C cases in the trace this works out to accelerating 78% of the C2Cs!
- What are some of the ways to implement the C2C improvement?



Some Alternatives to going after converting C2Cs to 2-hop latencies.

- Determining cache lines to hold in LP table:
  - All ownership operations
  - Add directory state bits to find lines that have history of C2Cs. (ie P&D protocol)
    - Either in memory directory state, or just in directory-cache state.
- Processing information with LP:
  - Use ownership as a hint and send requests to both owner node and directory
  - Move ordering point from directory to owner LP for those lines cached in LP.
  - Mixed Broadcast/Directory Protocol where LP match specifies line treated as Broadcast.
- Updating LP tables at processor side of agent:
  - Broadcast ownership info (or line is now in Broadcast mode info) for all chosen cache lines
  - Learn nodes that are active (with either directory-cache or LM structure) and multicast updates only to the corresponding LPs.